Inference in Bayesian networks - MCMC

Chapter 14.5.2

Markov Chains

A Markov chain defines a probabilistic transition model $q(\mathbf{x} \rightarrow \mathbf{x}')$ over states $\mathbf{x}$:

- for all x: $\sum_{\mathbf{x'}} q(\mathbf{x} \rightarrow \mathbf{x}') = 1$

Temporal Dynamics:

$$P^{(t+1)}(X^{(t+1)} = x') = \sum_{\mathbf{x}} P^{(t)}(X^{(t)} = x) q(\mathbf{x} \rightarrow \mathbf{x}')$$

Stationary distribution

$\pi_t(\mathbf{x})$ = probability in state $\mathbf{x}$ at time t
$\pi_{t+1}(\mathbf{x}')$ = probability in state $\mathbf{x}'$ at time $t+1$

$P^{(t+1)}(\mathbf{x}') \approx P^{(t)}(\mathbf{x}') = \Sigma_{\mathbf{x}} P^{(t)}(x) q(\mathbf{x} \rightarrow \mathbf{x}')$

π_{t+1} in terms of π_t and $q(\mathbf{x} \rightarrow \mathbf{x}')$

$$\pi_{t+1}(\mathbf{x}') = \Sigma_{\mathbf{x}} \pi_t(\mathbf{x}) q(\mathbf{x} \rightarrow \mathbf{x}')$$

Stationary distribution: $\pi_t = \pi_{t+1} = \pi$

$$\pi(\mathbf{x}') = \Sigma_{\mathbf{x}} \pi(\mathbf{x}) q(\mathbf{x} \rightarrow \mathbf{x}') \qquad \text{for all } \mathbf{x}'$$

If π exists, it is unique (specific to $q(\mathbf{x} \rightarrow \mathbf{x}')$)

In equilibrium, expected "outflow" = expected "inflow"

Detailed balance

"Outflow" = "inflow" for each pair of states:

$$\pi(\mathbf{x})q(\mathbf{x} \rightarrow \mathbf{x}') = \pi(\mathbf{x}')q(\mathbf{x}' \rightarrow \mathbf{x}) \qquad \text{for all } \mathbf{x},\ \mathbf{x}'$$

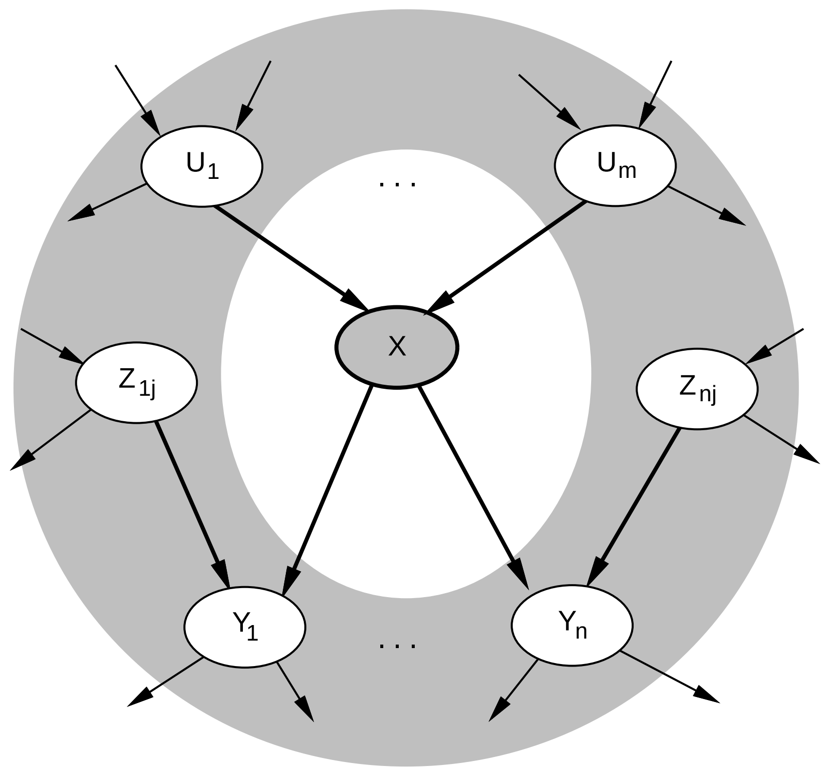
Detailed balance $\Rightarrow$ stationarity:

$$\begin{aligned}\Sigma_{\mathbf{x}}\pi(\mathbf{x})q(\mathbf{x} \rightarrow \mathbf{x}') &= \Sigma_{\mathbf{x}}\pi(\mathbf{x}')q(\mathbf{x}' \rightarrow \mathbf{x}) \\ &= \pi(\mathbf{x}')\Sigma_{\mathbf{x}}q(\mathbf{x}' \rightarrow \mathbf{x}) \\ &= \pi(\mathbf{x}')\end{aligned}$$

MCMC algorithms typically constructed by designing a transition probability q that is in detailed balance with desired π

Markov blanket

Each node is conditionally independent of all others given its
Markov blanket: parents + children + children's parents



Approximate inference using (MCMC)

Markov Chain Monte Carlo (MCMC)

Goal: compute $P(\mathbf{x} \in S)$
but P is too hard to sample from directly

Construct a Markov chain T whose unique stationary distribution is P

Sample $\mathbf{x}^{(0)}$ from some $P^{(0)}$ and generate $\mathbf{x}^{(t+1)}$ from $q(\mathbf{x}^t \rightarrow \mathbf{x}')$

Initially the samples far from distribution P. Use the samples only after the chain has run long enought to "mix"

Gibbs sampling

Gibbs sampling is a variant of Markov Chain Monte Carlo (MCMC)

Sample each variable in turn, given **all other variables**

Sampling X_i, let $\bar{\mathbf{X}}_i$ be all other nonevidence variables
Current values are x_i and $\bar{\mathbf{x}}_i$; $\mathbf{e}$ is fixed
Transition probability is given by

$$q(\mathbf{x} \rightarrow \mathbf{x}') = q(x_i, \bar{\mathbf{x}}_i \rightarrow x_i', \bar{\mathbf{x}}_i) = P(x_i'|\bar{\mathbf{x}}_i, \mathbf{e})$$

This gives detailed balance with true posterior $P(\mathbf{x}|\mathbf{e})$:

$$\begin{aligned}
\pi(\mathbf{x})q(\mathbf{x} \rightarrow \mathbf{x}') &= P(\mathbf{x}|\mathbf{e})P(x_i'|\bar{\mathbf{x}}_i, \mathbf{e}) = P(x_i, \bar{\mathbf{x}}_i|\mathbf{e})P(x_i'|\bar{\mathbf{x}}_i, \mathbf{e}) \\
&= P(x_i|\bar{\mathbf{x}}_i, \mathbf{e})P(\bar{\mathbf{x}}_i|\mathbf{e})P(x_i'|\bar{\mathbf{x}}_i, \mathbf{e}) \quad \text{(chain rule)} \\
&= P(x_i|\bar{\mathbf{x}}_i, \mathbf{e})P(x_i', \bar{\mathbf{x}}_i|\mathbf{e}) \qquad \text{(chain rule backwards)} \\
&= q(\mathbf{x}' \rightarrow \mathbf{x})\pi(\mathbf{x}') = \pi(\mathbf{x}')q(\mathbf{x}' \rightarrow \mathbf{x})
\end{aligned}$$

Approximate inference using Gibbs

"State" of network = current assignment to all variables.
Generate next state by sampling one variable given Markov blanket (mb)
Sample each variable in turn, keeping evidence fixed

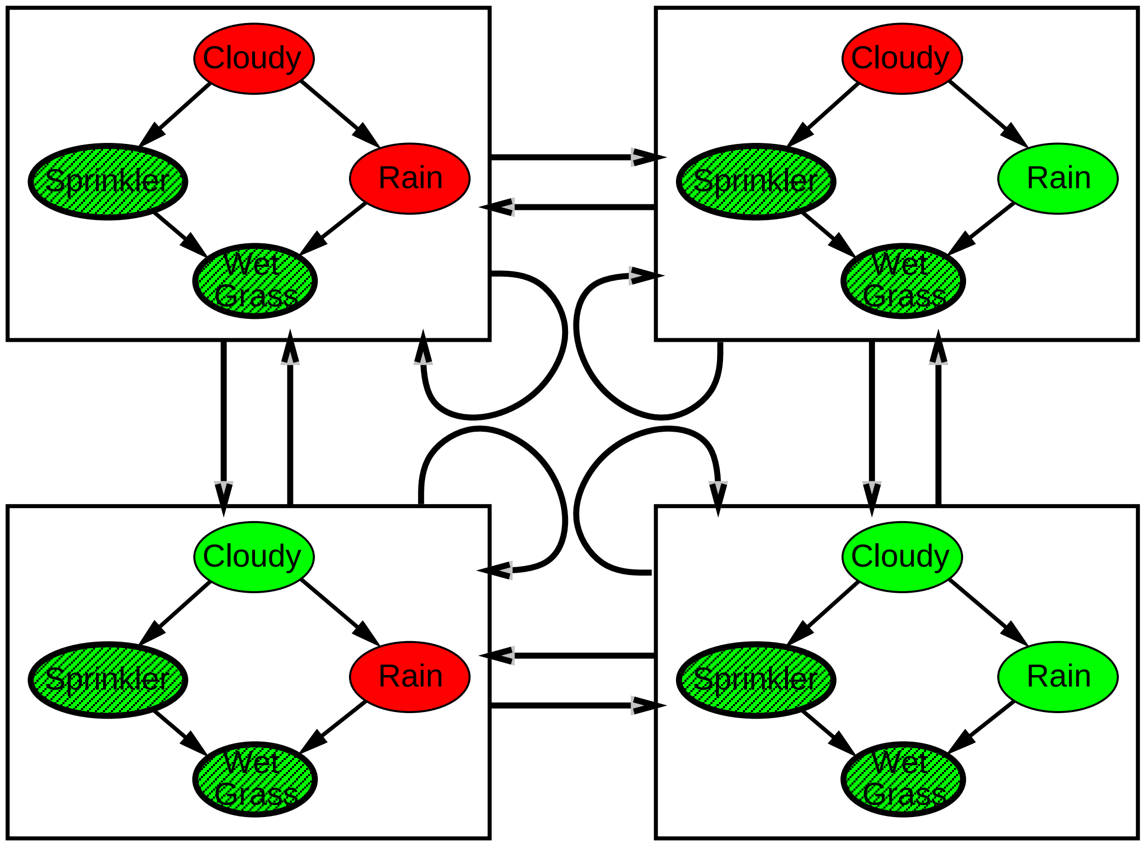
function GIBBS-ASK($X, \mathbf{e}, bn, N$) **returns** an estimate of $P(X|\mathbf{e})$
  local variables: $\mathbf{N}[X]$, a vector of counts over each value of X, initially zero
    $\mathbf{Z}$, the nonevidence variables in bn
    $\mathbf{x}$, the current state of the network, initially copied from $\mathbf{e}$

  initialize $\mathbf{x}$ with random values for the variables in $\mathbf{Z}$
  for $j = 1$ to N **do**
    for each Z_i in $\mathbf{Z}$ **do**
      set the value of Z_i in $\mathbf{x}$ by sampling from $\mathbf{P}(Z_i|mb(Z_i))$
        given the values of $MB(Z_i)$ in $\mathbf{x}$
      $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$ where x is the value of X in $\mathbf{x}$
  return NORMALIZE($\mathbf{N}[X]$)

This algorithm cycles through the variables, but choosing a variable to sample at random each time also works

The Markov chain

With $Sprinkler = true, WetGrass = true$, there are four states:



Wander about for a while, average what you see

Example contd.

Estimate $\mathbf{P}(Rain|Sprinkler = true, WetGrass = true)$

Sample $Cloudy$ or $Rain$ given its Markov blanket, repeat.
Count number of times $Rain$ is true and false in the samples.

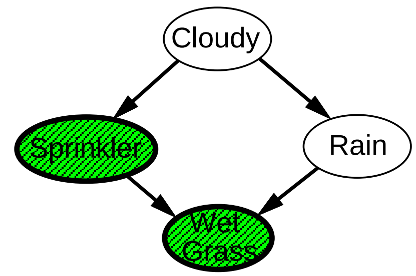
E.g., visit 100 states
  31 have $Rain = true$, 69 have $Rain = false$

$\hat{\mathbf{P}}(Rain|Sprinkler = true, WetGrass = true)$
  $= \textsc{Normalize}(\langle 31, 69 \rangle) = \langle 0.31, 0.69 \rangle$

Theorem: chain approaches stationary distribution:
  long-run fraction of time spent in each state is exactly
  proportional to its posterior probability

Markov blanket sampling

Markov blanket of $Cloudy$ is
$Sprinkler$ and $Rain$
Markov blanket of $Rain$ is
$Cloudy$, $Sprinkler$, and $WetGrass$



Probability given the Markov blanket is calculated as follows:
$P(x_i'|mb(X_i)) = P(x_i'|parents(X_i))\prod_{Z_j \in Children(X_i)} P(z_j|parents(Z_j))$

Easily implemented in message-passing parallel systems, brains

Main computational problems:
1) Difficult to tell if convergence has been achieved
2) Can be wasteful if Markov blanket is large:
$P(X_i|mb(X_i))$ won't change much (law of large numbers)

MCMC analysis: Outline

Transition probability $q(\mathbf{x} \rightarrow \mathbf{x}')$

Occupancy probability $\pi_t(\mathbf{x})$ at time t

Equilibrium condition on π_t defines stationary distribution $\pi(\mathbf{x})$
Note: stationary distribution depends on choice of $q(\mathbf{x} \rightarrow \mathbf{x}')$

Pairwise detailed balance on states guarantees equilibrium

Gibbs sampling transition probability:
sample each variable given current values of all others
$\Rightarrow$ detailed balance with the true posterior

For Bayesian networks, Gibbs sampling reduces to
sampling conditioned on each variable's Markov blanket

Performance of approximation algorithms

Absolute approximation: $|P(X|\mathbf{e}) - \hat{P}(X|\mathbf{e})| \leq \epsilon$

Relative approximation: $\frac{|P(X|\mathbf{e}) - \hat{P}(X|\mathbf{e})|}{P(X|\mathbf{e})} \leq \epsilon$

Relative $\Rightarrow$ absolute since $0 \leq P \leq 1$ (may be $O(2^{-n})$)

Randomized algorithms may fail with probability at most δ

Polytime approximation: $\mathsf{poly}(n, \epsilon^{-1}, \log \delta^{-1})$

Theorem (Dagum and Luby, 1993): both absolute and relative approximation for either deterministic or randomized algorithms are NP-hard for any $\epsilon, \delta < 0.5$

(Absolute approximation polytime with no evidence—Chernoff bounds)

Summary

Exact inference by variable elimination:

- polytime on polytrees, NP-hard on general graphs
- space = time, very sensitive to topology

Approximate inference by LW, MCMC:

- LW does poorly when there is lots of (downstream) evidence
- LW, MCMC generally insensitive to topology
- Convergence can be very slow with probabilities close to 1 or 0
- Can handle arbitrary combinations of discrete and continuous variables